

# 크롤링

```
def parse_url(url, css_selector):  
    r = requests.get(url)  
    soup = BeautifulSoup(r.content, 'lxml')  
    s = soup.select_one(css_selector)  
    with open('article.txt', 'w+') as f:  
        f.write(s.text.strip())  
    return f.name
```

소프트웨어융합대학원  
진혜진

# 크롤링

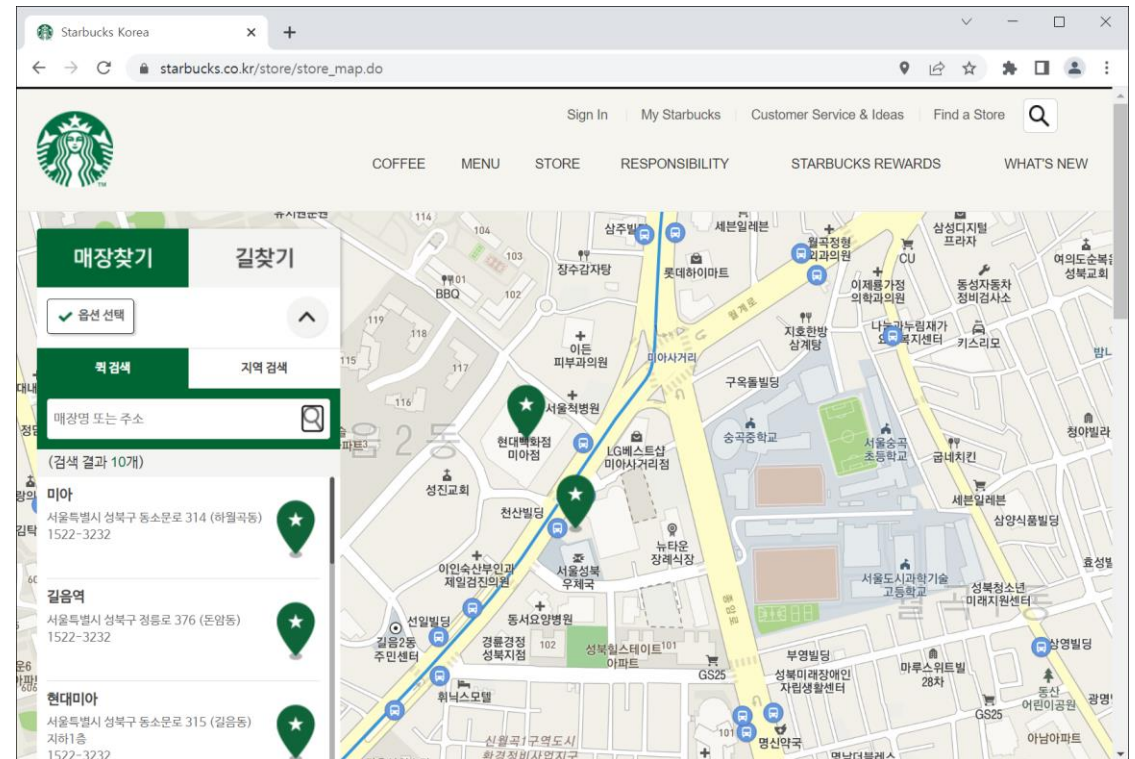
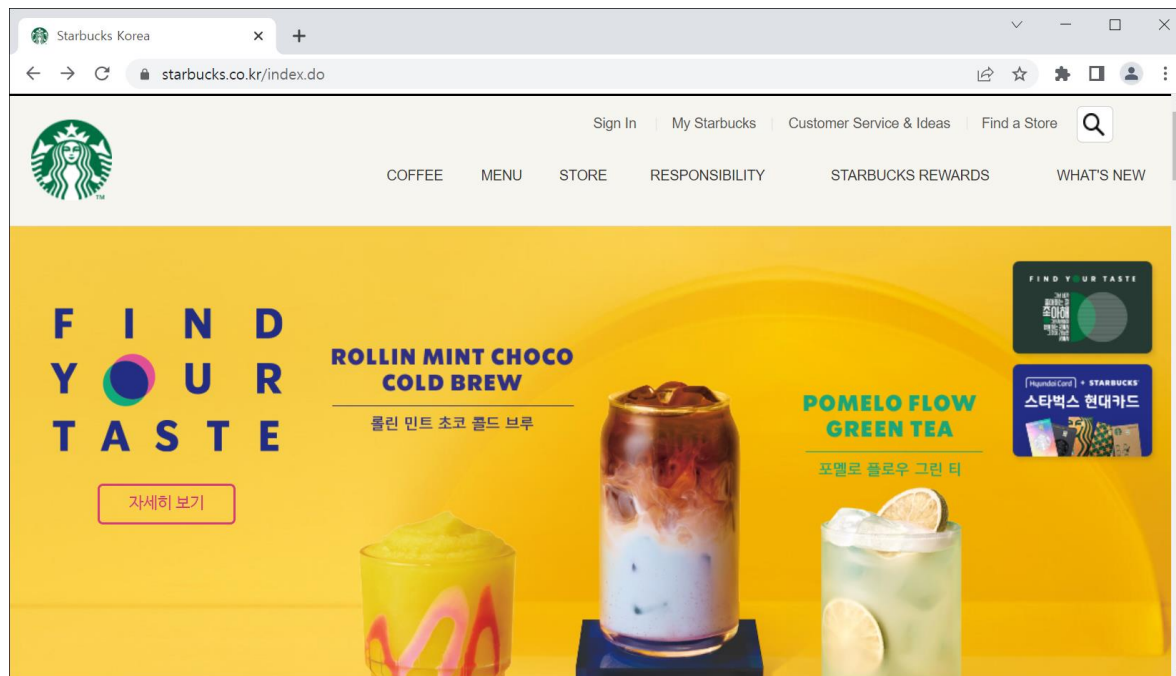
## 1. 웹 크롤링

- 웹 페이지에 있는 정보를 가지고 오는 것을 의미함
- selenium 라이브러리의 webdriver를 활용해 웹 브라우저를 조작하고, BeautifulSoup 라이브러리를 활용해 웹 페이지 상의 HTML 데이터에서 필요한 정보를 가져옴
- Selenium의 webdriver는 크롬이나 익스플로러 등에서 사이트 접속, 버튼 클릭 등 웹 브라우저에서 사람이 할 수 있는 일들을 코드를 통해서 제어할 수 있는 라이브러리임
- webdriver를 활용하기 위해서는 웹 브라우저의 종류에 따라 제어하는 드라이버가 필요함
  - 크롬 드라이버를 이용해 크롤링을 진행함

# 데이터 수집

## 1. 크롤링

- 서울시 스타벅스 매장 목록 데이터 생성



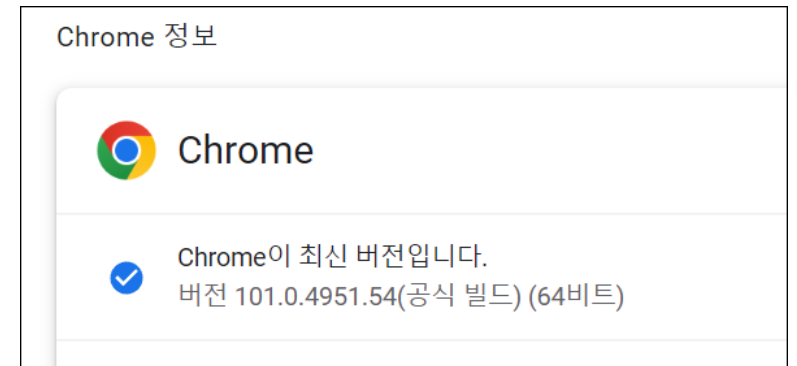
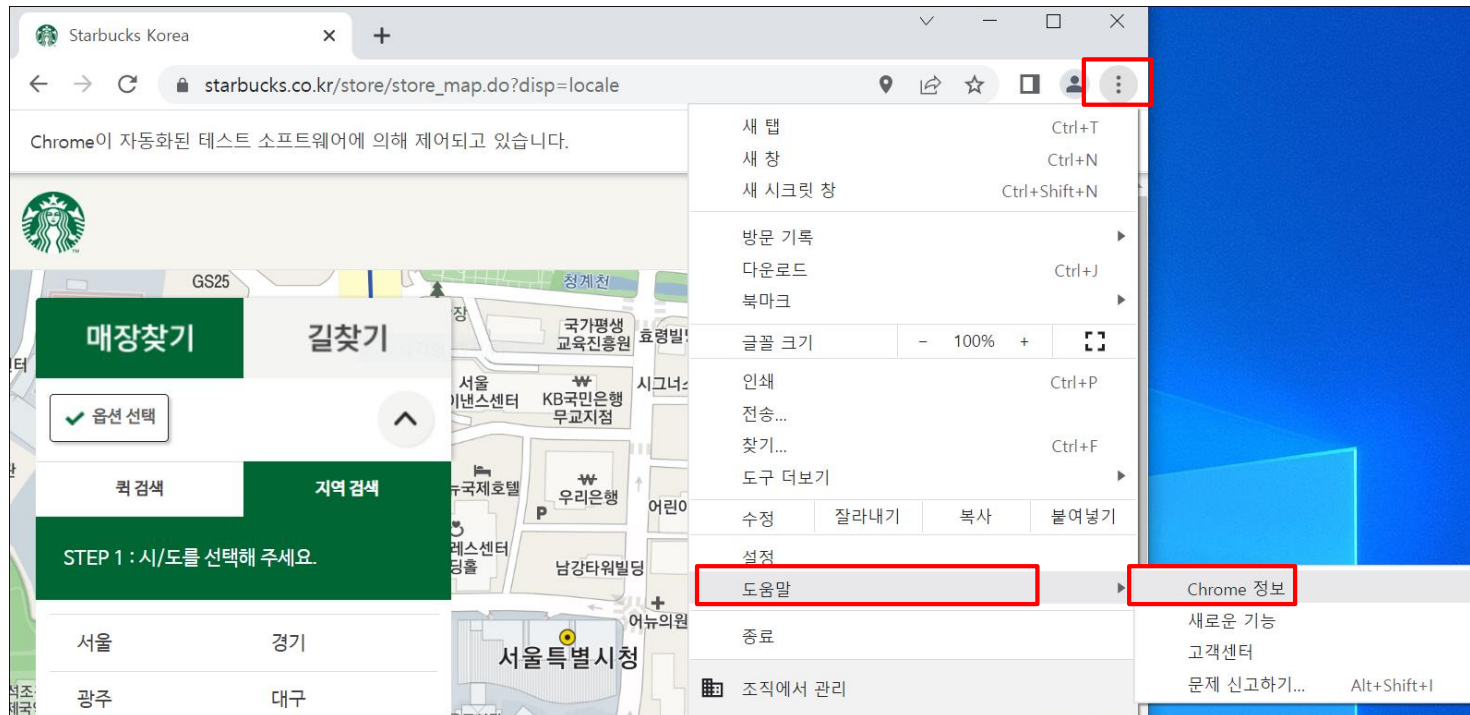
# 크롤링

## 1. selenium과 크롬 드라이버 설치

- `from selenium import webdriver`
- 설치되지 않았다는 에러 메시지가 출력되면 `!pip install selenium==3.141` 코드를 실행해 selenium 설치함
- 크롬 드라이버는 selenium의 webdriver를 통해 파이썬에서 크롬 브라우저를 제어할 수 있게 해줌
  - 기존에 설치되어 있더라도 selenium으로 작동하는 크롬드라이버는 별도의 파일이 필요하기 때문에 다운로드 받아야 함

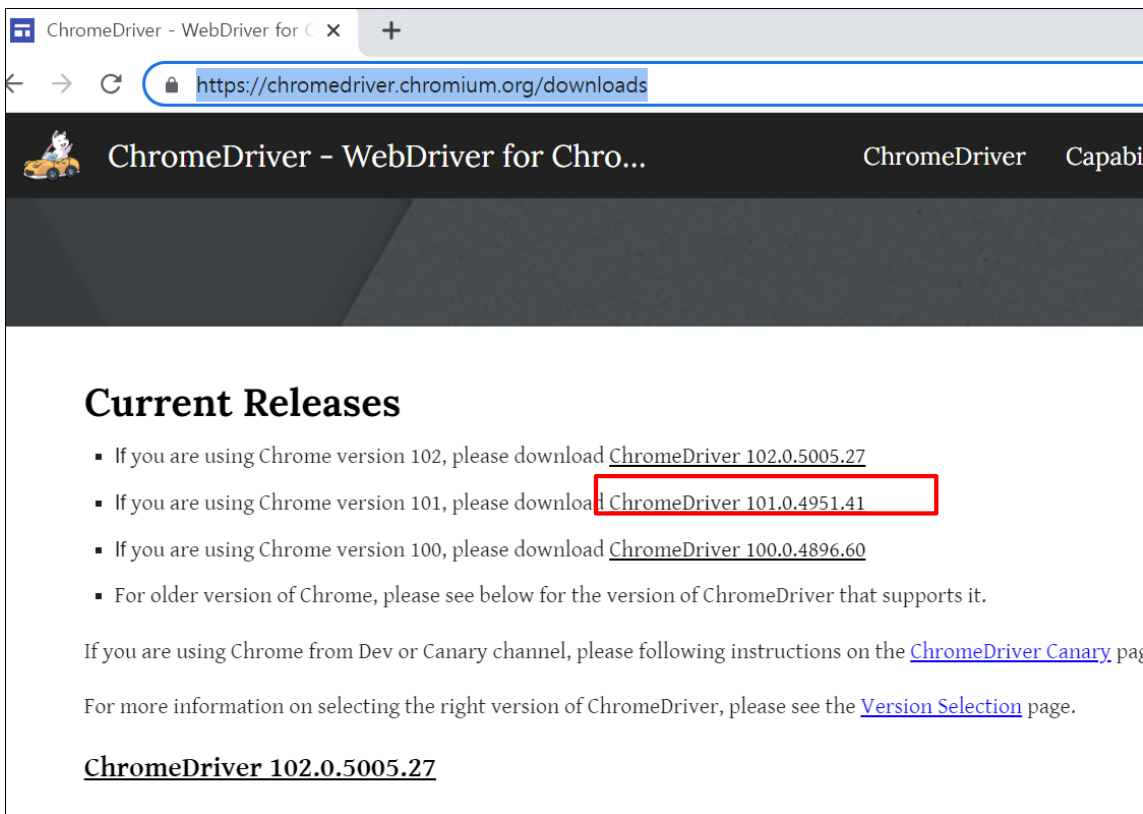
# 크롤링

- 현재 사용 중인 크롬 버전 확인



# 크롤링

- <https://chromedriver.chromium.org/downloads>



The screenshot shows the ChromeDriver download page. The browser's address bar displays the URL <https://chromedriver.chromium.org/downloads>. The page title is "ChromeDriver - WebDriver for Chrome". Under the "Current Releases" section, there is a list of instructions for different Chrome versions. The instruction for Chrome version 101 is highlighted with a red box, showing "ChromeDriver 101.0.4951.41". Below this, there is a link to the "Version Selection" page.

**Current Releases**




- If you are using Chrome version 102, please download [ChromeDriver 102.0.5005.27](#)
- If you are using Chrome version 101, please download [ChromeDriver 101.0.4951.41](#)
- If you are using Chrome version 100, please download [ChromeDriver 100.0.4896.60](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

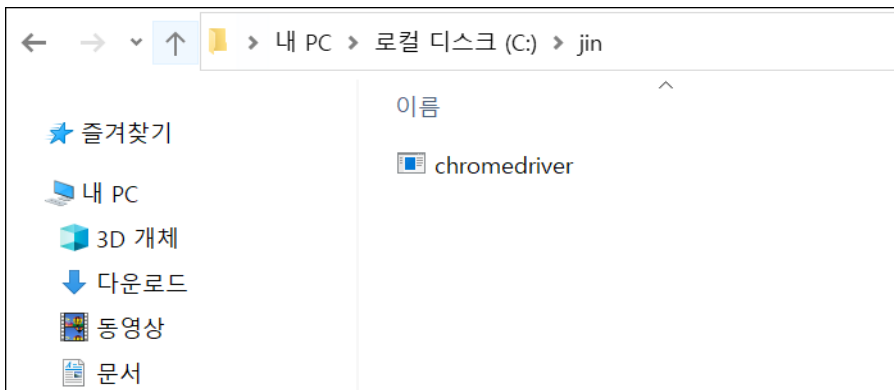
If you are using Chrome from Dev or Canary channel, please following instructions on the [ChromeDriver Canary](#) page.

For more information on selecting the right version of ChromeDriver, please see the [Version Selection](#) page.

**[ChromeDriver 102.0.5005.27](#)**

### Index of /101.0.4951.41/

|   | Name                                      | Last modified       | Size   |
|---|---|---------------------|--------|
|  | <a href="#">Parent Directory</a>          |                     | -      |
|  | <a href="#">chromedriver linux64.zip</a>  | 2022-04-27 07:02:29 | 5.92MB |
|  | <a href="#">chromedriver mac64.zip</a>    | 2022-04-27 07:02:31 | 7.88MB |
|  | <a href="#">chromedriver mac64_m1.zip</a> | 2022-04-27 07:02:34 | 7.19MB |
|  | <a href="#">chromedriver win32.zip</a>    | 2022-04-27 07:02:37 | 6.05MB |
|  | <a href="#">notes.txt</a>                 | 2022-04-27 07:02:42 | 0.00MB |



The screenshot shows a Windows File Explorer window. The address bar displays the path "내 PC > 로컬 디스크 (C:) > jin". The left sidebar shows the "즐거찾기" (QuickTime) section with "내 PC" (This PC) selected. The main area shows a folder named "chromedriver".

# 크롤링

---

- `driver.get(url)` 을 통해 특정 URL에 접속 할 수 있음
- 사이트에 접속하거나 클릭하는 등의 작업을 한 뒤에는 해당 페이지의 정보를 다 받을 때 까지 대기 시간이 필요함



# 크롤링

크롤링을 이용한 서울시 스타벅스 매장 목록 데이터 생성

1. 크롤링 실행하기 위해 selenium 라이브러리의 webdriver 불러옴
2. 크롤링으로 가져온 HTML에서 정보를 추출하기 위해 BeautifulSoup 라이브러리 추가
3. 추출된 데이터를 엑셀로 저장하기 위해 pandas라이브러리 추가

```
1 # 라이브러리 импорт
2 !pip install selenium==3.141
3
4 from selenium import webdriver
5 from bs4 import BeautifulSoup
6 import pandas as pd
```

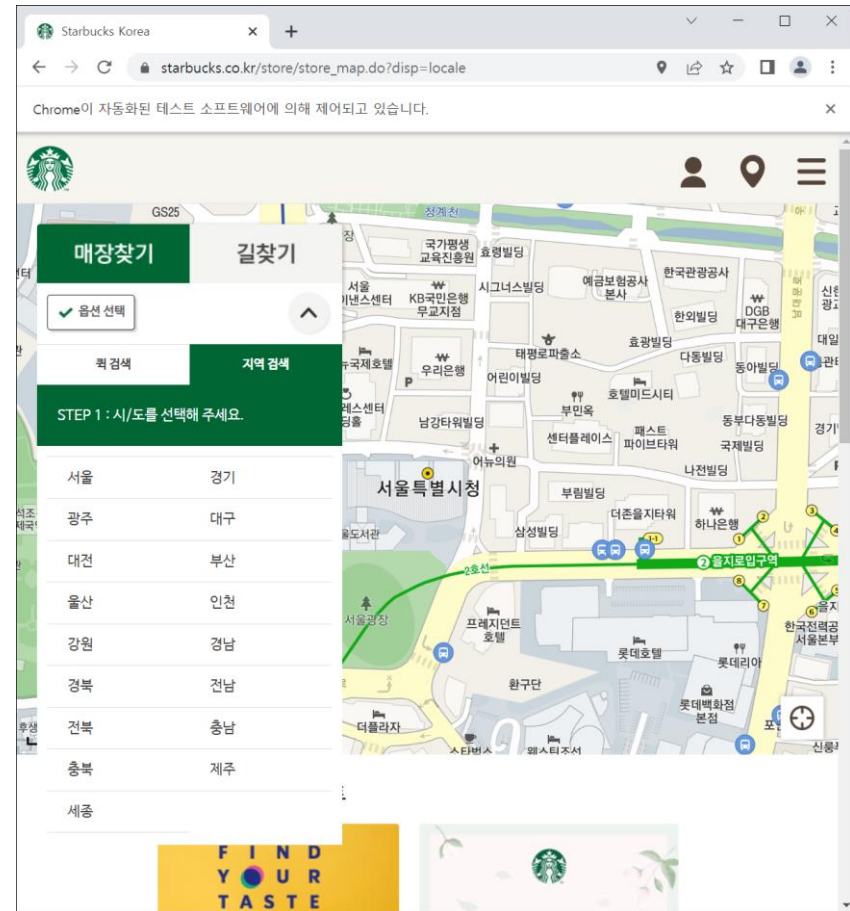
Requirement already satisfied: selenium==3.141 in c:\wprogramdata\Wanaconda3\lib\site-packages  
Requirement already satisfied: urllib3 in c:\wprogramdata\Wanaconda3\lib\site-packages

webdriver.Chrome함수를 이용해 크롬 브라우저 실행 후 driver 변수에 할당

driver.get(url) 명령어를 통해 해당 url로 이동

크롬 브라우저가 자동으로 실행됨

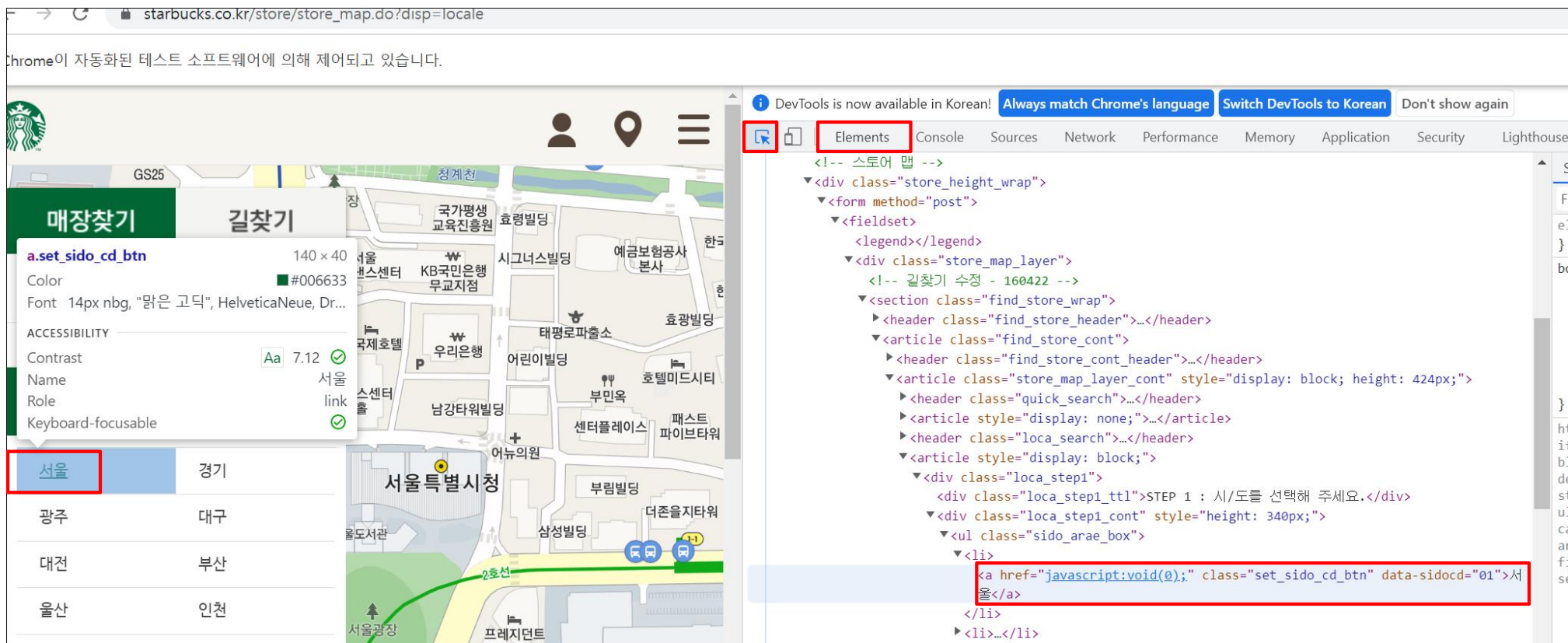
```
1 #webdriver 실행 후 스타벅스의 지역별 매장 검색 화면에 접속
2 driver = webdriver.Chrome('c:/jin/chromedriver.exe')
3 # driver = webdriver.Chrome('./chromedriver')
4 url = 'https://www.istarbucks.co.kr/store/store_map.do?disp=locale'
5 driver.get(url)
```





# 크롤링

## 1. 크롬 브라우저에서 서울 버튼을 위치를 얻기 위해 개발자 도구 실행(단축기 F12)



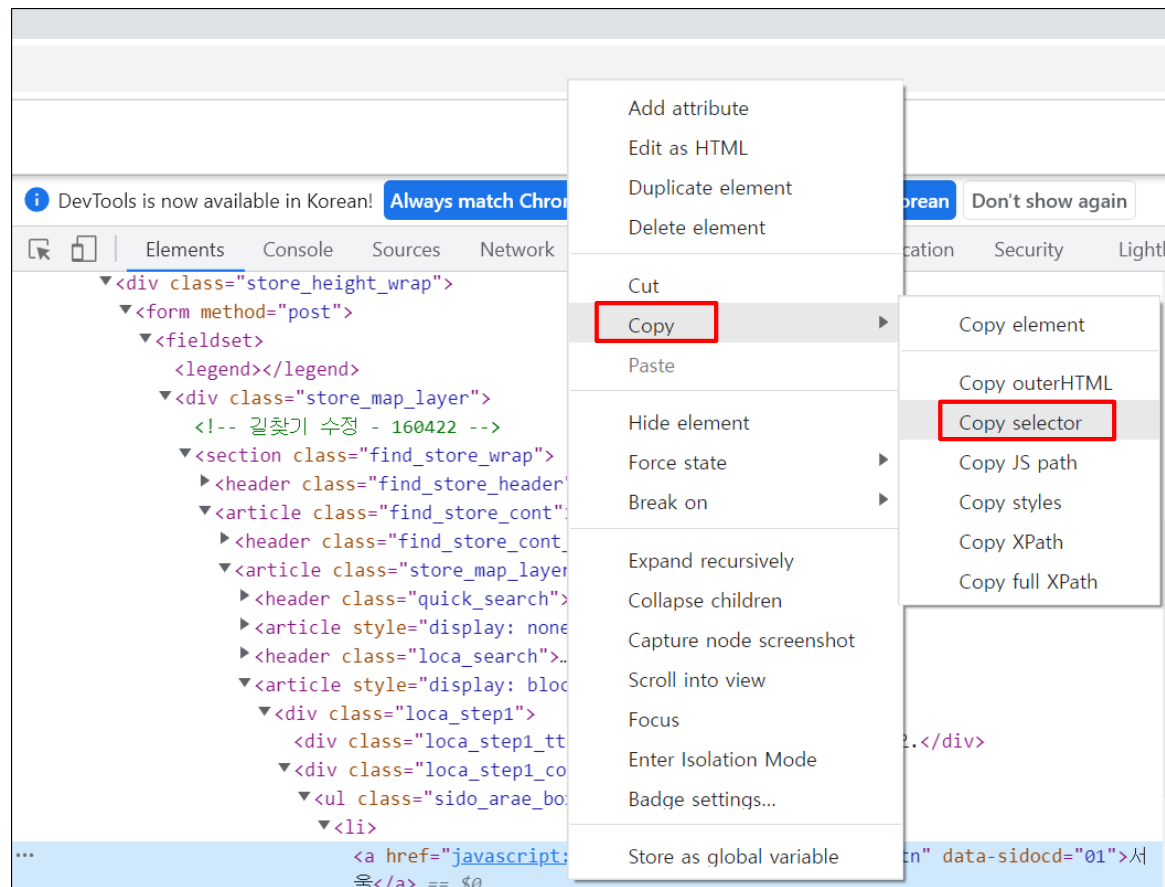
The screenshot shows the Starbucks website's location selection interface. The '매장찾기' (Find Store) section is active, displaying a list of locations. The '서울' (Seoul) button is highlighted in the list. The DevTools Elements panel is open, showing the HTML structure of the page. The 'set\_sido\_cd\_btn' button is highlighted in the HTML code.

HTML structure (Elements panel):

```
<!-- 스토어 맵 -->
<div class="store_height_wrap">
  <form method="post">
    <fieldset>
      <legend></legend>
      <div class="store_map_layer">
        <!-- 길찾기 수정 - 160422 -->
        <section class="find_store_wrap">
          <header class="find_store_header">...</header>
          <article class="find_store_cont">
            <header class="find_store_cont_header">...</header>
            <article class="store_map_layer_cont" style="display: block; height: 424px;">
              <header class="quick_search">...</header>
              <article style="display: none;">...</article>
              <header class="loca_search">...</header>
              <article style="display: block;">
                <div class="loca_step1">
                  <div class="loca_step1_ttl">STEP 1 : 시/도를 선택해 주세요.</div>
                  <div class="loca_step1_cont" style="height: 340px;">
                    <ul class="sido_arae_box">
                      <li>
                        <a href="javascript:void(0);" class="set_sido_cd_btn" data-sidocd="01">서울</a>
                      </li>
                    </ul>
                  </div>
                </div>
              </article>
            </article>
          </section>
        </div>
      </div>
    </fieldset>
  </form>
</div>
```

# 크롤링

- 하이라이트 표시된 태그 영역에서 Copy selector를 클릭하면 전체 HTML 문서 내에서 '서울' 버튼에 해당하는 태그의 구조 정보(CSS selector)가 복사됨
- `#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_step1 > div.loca_step1_cont > ul > li:nth-child(1) > a`

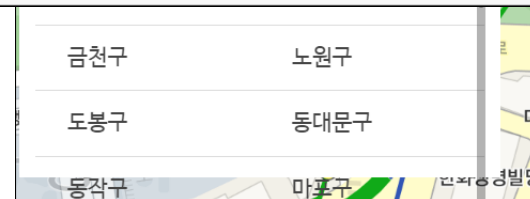
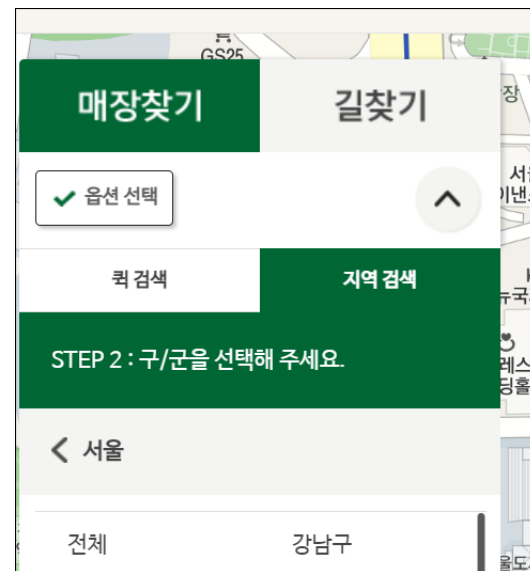


# 크롤링

- `find_element_by_css_selector(seoul_btn)`으로 `seoul_btn`에 저장된 '서울' 버튼 위치를 선택하고 `click()`으로 해당 버튼 클릭함
- 서울시 구/군 목록이 나타나고 구/군별 스타벅스 매장을 조회할 수 있는 화면이 나타남
- `#webdriver`로 '서울' 버튼 요소를 찾아 클릭

*#webdriver로 '서울' 버튼 요소를 찾아 클릭*

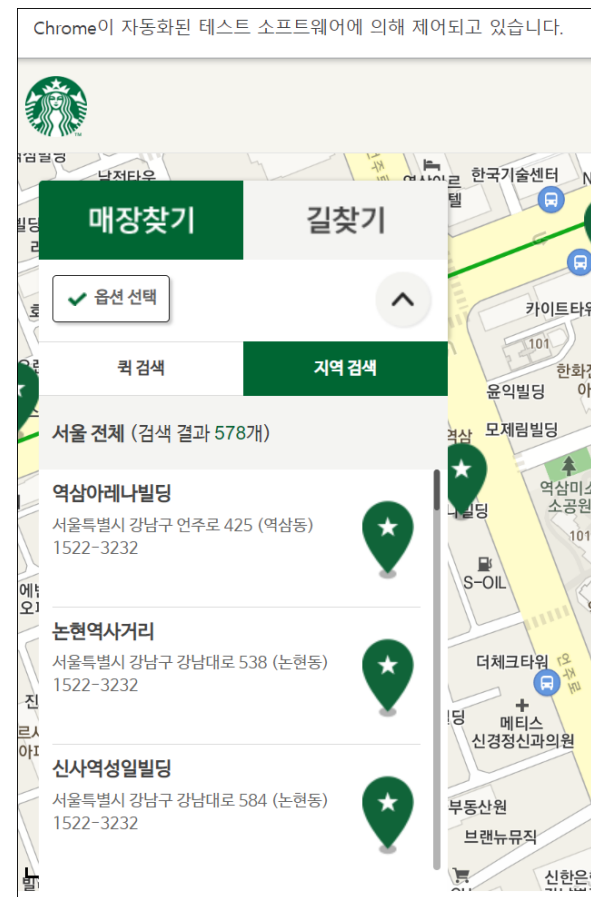
```
seoul_btn = '#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_step1 > div.loca_step1_cont > ul > li:nth-child(1) > a'  
driver.find_element_by_css_selector(seoul_btn).click()
```



# 크롤링

- 서울시 전체 스타벅스 매장 목록이 나타남

```
# webdriver로 '전체' 버튼 요소를 찾아 클릭  
all_btn = '#mCSB_2_container > ul > li:nth-child(1) > a'  
driver.find_element_by_css_selector(all_btn).click()
```



# 크롤링

- `driver.page_source`를 통해 크롬 브라우저에 현재 화면에 나타난 웹 페이지의 HTML을 가져올 수 있다.
- `html.parser`는 HTML 문법을 이해하고 웹 페이지의 정보를 분류하는 역할을 함

```
1 # BeautifulSoup으로 HTML 파서 만들기
2 html = driver.page_source
3 soup = BeautifulSoup(html, 'html.parser')
```

```
1 # select()를 이용해 원하는 HTML 태그를 모두 찾아오기
2 starbucks_soup_list = soup.select('li.quickResultLstCon')
3 print(len(starbucks_soup_list))
```

578

# 크롤링

- 태그 내 매장 상세 정보 확인하기 위해 리스트의 첫 번째 원소인 `starbucks_soup_list[0]`의 데이터를 살펴봄
- 하나의 `<li>` 태그가 출력됨

```
1 # 태그 문자열 살펴보기
2 starbucks_soup_list[0]
3
```

```
<li class="quickResultLstCon" data-code="3762" data-hlytag="null" data-index="0" data-lat="37.501087" data-
삼아레나빌딩" data-storecd="1509" style="background:#fff"> <strong data-my_siren_order_store_yn="N" data-na
="1509" data-yn="N">역삼아레나빌딩 </strong> <p class="result_details">서울특별시 강남구 언주로 425 (역삼동
in_general">리저브 매장 2번</i></li>
```

# 크롤링

- 스타벅스 매장 정보 샘플 확인

```
1 # 스타벅스 매장 정보 샘플 확인
2 starbucks_store = starbucks_soup_list[0]
3 name = starbucks_store.select('strong')[0].text.strip()
4 lat = starbucks_store['data-lat'].strip()
5 lng = starbucks_store['data-long'].strip()
6 store_type = starbucks_store.select('i')[0]['class'][0][4:]
7 address = str(starbucks_store.select('p.result_details')[0]).split('<br />')[0].split('>')[1]
8 tel = str(starbucks_store.select('p.result_details')[0]).split('<br />')[1].split('<')[0]
9
10 print(name)          # 매장명
11 print(lat)           # 위도
12 print(lng)           #경도
13 print(store_type)    # 매장 타입
14 print(address)       # 주소
15 print(tel)           # 전화번호
```

역삼아레나빌딩  
37.501087  
127.043069  
general  
서울특별시 강남구 언주로 425 (역삼동)  
1522-3232



# 크롤링

## ■ 서울시 스타벅스 매장 목록 데이터 만들기

```
1 # 서울시 스타벅스 매장 목록 데이터 만들기
2 starbucks_list = []
3 for item in starbucks_soup_list:
4     name = item.select('strong')[0].text.strip()
5     lat = item['data-lat'].strip()
6     lng = item['data-long'].strip()
7     store_type = item.select('i')[0]['class'][0][4:]
8     address = str(item.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
9     tel = str(item.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]
10
11     starbucks_list.append( [ name, lat, lng, store_type, address, tel])
```

```
1 # pandas의 데이터프레임 생성
2 columns = ['매장명', '위도', '경도', '매장타입', '주소', '전화번호']
3 seoul_starbucks_df = pd.DataFrame(starbucks_list, columns = columns)
4 seoul_starbucks_df.head()
```

|   | 매장명     | 위도        | 경도         | 매장타입    | 주소                         | 전화번호      |
|---|---------|-----------|------------|---------|----------------------------|-----------|
| 0 | 역삼아레나빌딩 | 37.501087 | 127.043069 | general | 서울특별시 강남구 언주로 425 (역삼동)    | 1522-3232 |
| 1 | 논현역사거리  | 37.510178 | 127.022223 | general | 서울특별시 강남구 강남대로 538 (논현동)   | 1522-3232 |
| 2 | 신사역성일빌딩 | 37.514132 | 127.020563 | general | 서울특별시 강남구 강남대로 584 (논현동)   | 1522-3232 |
| 3 | 국기원사거리  | 37.499517 | 127.031495 | general | 서울특별시 강남구 테헤란로 125 (역삼동)   | 1522-3232 |
| 4 | 대치재경빌딩R | 37.494668 | 127.062583 | reserve | 서울특별시 강남구 남부순환로 2947 (대치동) | 1522-3232 |

- 데이터프레임의 요약 정보 확인

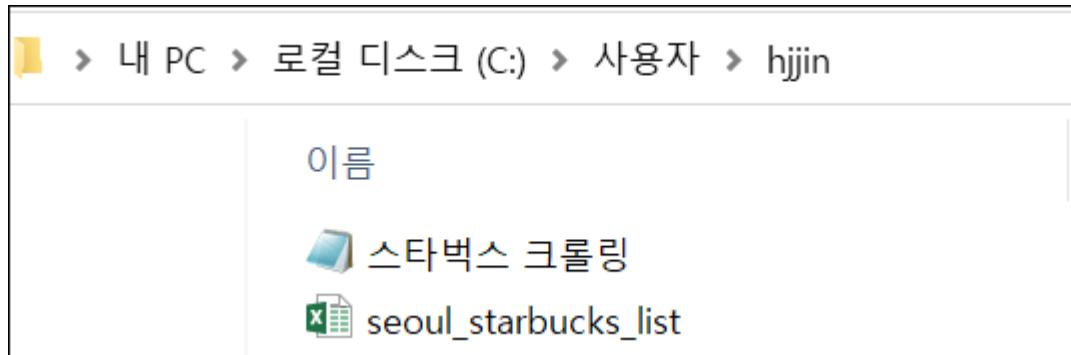
```
1 # 데이터프레임의 요약 정보 확인
2 seoul_starbucks_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 578 entries, 0 to 577
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   매장명      578 non-null    object
1   위도         578 non-null    object
2   경도         578 non-null    object
3   매장타입     578 non-null    object
4   주소         578 non-null    object
5   전화번호     578 non-null    object
dtypes: object(6)
memory usage: 27.2+ KB
```

# 크롤링

## ■ 엑셀로 저장

```
1 # 엑셀로 저장
2 seoul_starbucks_df.to_excel('seoul_starbucks_list.xlsx', index=False)
```



|    | A        | B          | C           | D       | E                          | F         |
|----|----------|------------|-------------|---------|----------------------------|-----------|
| 1  | 매장명      | 위도         | 경도          | 매장타입    | 주소                         | 전화번호      |
| 2  | 역삼아레나빌딩  | 37.501087  | 127.043069  | general | 서울특별시 강남구 언주로 425 (역삼동)    | 1522-3232 |
| 3  | 논현역사거리   | 37.510178  | 127.022223  | general | 서울특별시 강남구 강남대로 538 (논현동)   | 1522-3232 |
| 4  | 신사역성일빌딩  | 37.514132  | 127.020563  | general | 서울특별시 강남구 강남대로 584 (논현동)   | 1522-3232 |
| 5  | 국기원사거리   | 37.499517  | 127.031495  | general | 서울특별시 강남구 테헤란로 125 (역삼동)   | 1522-3232 |
| 6  | 대치재경빌딩R  | 37.494668  | 127.062583  | reserve | 서울특별시 강남구 남부순환로 2947 (대치동) | 1522-3232 |
| 7  | 봉은사역     | 37.515000  | 127.063196  | general | 서울특별시 강남구 봉은사로 619 (삼성동)   | 1522-3232 |
| 8  | 압구정윤성빌딩  | 37.5227934 | 127.0286009 | general | 서울특별시 강남구 논현로 834 (신사동)    | 1522-3232 |
| 9  | 코엑스별마당   | 37.510150  | 127.060275  | general | 서울특별시 강남구 영동대로 513 (삼성동)   | 1522-3232 |
| 10 | 삼성역삼유센터R | 37.507750  | 127.060651  | reserve | 서울특별시 강남구 테헤란로 518 (대치동)   | 1522-3232 |
| 11 | 압구정R     | 37.5273669 | 127.033061  | reserve | 서울특별시 강남구 언주로 861 (신사동)    | 1522-3232 |
| 12 | 수서역R     | 37.488008  | 127.102650  | reserve | 서울특별시 강남구 광평로 281 (수서동)    | 1522-3232 |
| 13 | 양재강남빌딩R  | 37.485192  | 127.036685  | reserve | 서울특별시 강남구 남부순환로 2621 (도곡동) | 1522-3232 |
| 14 | 선릉동신빌딩R  | 37.505321  | 127.050409  | reserve | 서울특별시 강남구 테헤란로 409 (삼성동)   | 1522-3232 |
| 15 | 봉은사로선정릉  | 37.511293  | 127.048409  | general | 서울특별시 강남구 봉은사로 446 (삼성동)   | 1522-3232 |
| 16 | 강남오거리    | 37.502117  | 127.026672  | general | 서울특별시 강남구 봉은사로2길 39 (역삼동)  | 1522-3232 |
| 17 | 스타필드코엑스몰 | 37.50999   | 127.061455  | reserve | 서울특별시 강남구 영동대로 513 (삼성동)   | 1522-3232 |
| 18 | 강남구청정문   | 37.518181  | 127.045995  | general | 서울특별시 강남구 학동로 419 (청담동)    | 1522-3232 |
| 19 | 도곡공원     | 37.492805  | 127.041309  | general | 서울특별시 강남구 도곡로 205 (역삼동)    | 1522-3232 |
| 20 | 강남R      | 37.497711  | 127.028439  | reserve | 서울특별시 강남구 강남대로 390 (역삼동)   | 1522-3232 |
| 21 | 대치역마사거리  | 37.498973  | 127.060172  | general | 서울특별시 강남구 도곡로 457 (대치동)    | 1522-3232 |
| 22 | 청담영동대로   | 37.522156  | 127.056449  | general | 서울특별시 강남구 영동대로 720 (청담동)   | 1522-3232 |
| 23 | 압구정      | 37.526283  | 127.02956   | general | 서울특별시 강남구 압구정로30길 17 (신사동) | 1522-3232 |
| 24 | 신사거리스    | 37.521922  | 127.022521  | general | 서울특별시 강남구 가로수길 59          | 1522-3232 |